AMENDMENTS TO THE CLAIMS:

1.   (Currently Amended) A method, comprising:

(a) creating in a design environment a file that determines a metadata that relates at least one business object and at least one query;

(b) communicating the file to a mobile device;

(c) storing the file on the mobile device;

(d) transforming the file into a binary structure at an initial run of a computer application running on the mobile device, the binary structure adapted to be read by the computer application; and

(e) recording the binary structure in a memory of the mobile device, wherein the method further comprises:

storing a datestamp and a filesize in the memory with the binary structure, the datestamp and the filesize uniquely identifying the file corresponding to the binary structure;

determining whether the file has been updated upon a startup of the computer application by comparing a ~~further~~ datestamp and a ~~further~~ filesize of a ~~further~~ file ~~stored~~ currently loaded on the mobile device to the datestamp and the filesize, respectively, recorded with the binary structure; and

mapping ~~the~~ a binary structure, transformed from the currently loaded file, by the computer application ~~from~~into the memory if the file has not been updated.

2.   (Original) The method of claim 1, wherein the method is adapted to create a database access system.

3.   (Original) The method of claim 1, wherein the file is an XML file.

4.   (Canceled)

5.   (Canceled)

6.   (Canceled)

7.   (Currently Amended) The method of claim 1, wherein, if the ~~further~~ datestamp and the ~~further~~ filesize of the ~~further~~ file ~~stored~~ currently loaded on the mobile device correspond with the datestamp and the filesize recorded with the binary structure, then the file and the further file are identical and the file has not been updated.

8.   (Currently Amended) The method of claim 1, wherein if the ~~further~~ datestamp and the ~~further~~ filesize of the ~~further~~ file ~~stored~~ currently loaded on the mobile device do not correspond with the datestamp and the filesize recorded with the binary structure when compared, then the file and the further file are not identical and the file has ~~not~~ been updated.

9.   (Original) The method of claim 8, further comprising repeating operations (d) and (e) for the further file.

10.   (Original) The method of claim 1, further comprising:
            creating a new file to change the metadata; and
            repeating the operations of (b), (c), (d), and (e) for the new file.

11.   (Original) The method of claim 1, wherein the metadata describes at least one of:
            actual data;
            at least one attribute for the at least one business object;
            at least one relationship between a plurality of business objects; and
            the at least one query.

12.   (Original) The method of claim 1, wherein the mobile device includes at least one of a laptop computer and a personal digital assistant.

13.   (Original) The method of claim 1, wherein the recording of the binary structure in the memory includes mapping the binary structure into an address space.

14.   (Original) The method of claim 1, further comprising mapping the binary structure for a subsequent run of the computer application until the file is updated with a new file.

15.   (Currently Amended) A method, comprising:

receiving at a mobile device a file that includes a metadata that relates at least one object and at least one record for a computer application;

storing the file on the mobile device;

converting the file into a binary file at a first running of the computer application and storing the binary file on the mobile device;

storing a datestamp and a filesize uniquely identifying the file in a memory of the mobile device;

determining whether the file has been updated upon ~~the first~~ a subsequent running of the computer application by comparing a ~~further~~ datestamp and a ~~further~~ filesize of a ~~further~~ file ~~stored~~ currently loaded on the mobile device to the stored datestamp and the stored filesize, respectively; and

mapping ~~the~~ a binary file, converted from the currently loaded file, by the computer application ~~from~~into the memory if the file has not been updated.

16. (Previously Presented) The method of claim 15, further comprising recording the binary file in the memory of the mobile device.

17. (Currently Amended) The method of claim 16, further comprising reading the binary ~~structure~~ file at a second running of the computer application.

18. (Original) The method of claim 15, wherein the file is an XML file.

19. (Canceled)

20. (Currently Amended) A system for updating a database access program, comprising:

a design server including a memory and a processor adapted to create a file that relate a plurality of objects in a computer application and records in a database; and

a mobile device including a processor, a program memory, a database memory, and a communication interface coupled to the network;

wherein the design server is adapted to create the file;

wherein the mobile device is adapted to transform the file into a binary structure at an initial run of a computer application running on the mobile device, the binary structure adapted to be read by the computer application;

wherein the mobile device stores a datestamp and a filesize uniquely identifying the file in a memory of the mobile device;

wherein the mobile device determines whether the file has been updated upon the initial run of the computer application by comparing a ~~further~~ datestamp and a ~~further~~ filesize of a ~~further~~ file ~~stored~~ currently loaded on the mobile device to the stored datestamp and the stored filesize, respectively; and

wherein the mobile device maps ~~the~~ a binary file, transformed from the currently loaded file, by the computer application ~~from~~ into the memory if the file has not been updated.

21.   (Canceled)

22.   (Currently Amended) The method of claim 15, wherein, if the ~~further~~ datestamp and the ~~further~~ filesize of the ~~further~~ file ~~stored~~ currently loaded on the mobile device correspond with the stored datestamp and the stored filesize, then the file and the ~~further~~currently loaded file are identical and the file has not been updated.

23.   (Currently Amended) The method of claim 15, wherein if the ~~further~~ datestamp and the ~~further~~ filesize of the ~~further~~ file ~~stored~~ currently loaded on the mobile device do not correspond with the stored datestamp and the stored filesize recorded with the binary structure, then the file and the ~~further~~ currently loaded file are not identical and the file has been updated.

24.   (Currently Amended) The system of claim 20, wherein, if the ~~further~~ datestamp and the ~~further~~ filesize of the ~~further~~ file ~~stored~~ currently loaded on the mobile device correspond with the datestamp and the filesize, then the file and the further file are identical and the file has not been updated.

25.   (Currently Amended) The system of claim 20, wherein if the ~~further~~ datestamp and the ~~further~~ filesize of the ~~further~~ file ~~stored~~ currently loaded on the mobile device do not correspond with the datestamp and the filesize recorded with the binary structure, then the file and the ~~further~~ currently loaded file are not identical and the file has been updated.

26.   (Previously Presented) A method, comprising:

starting an application in a run-time environment at a mobile device;

receiving at the mobile device a first file created in a development environment at a remote device;

converting the first file to a first binary structure;

storing a first filesize and a first datestamp of the first file with the first binary structure in a memory of the mobile device;

receiving at the mobile device a second file created in the development environment at the remote device;

determining if the second file corresponds to the stored first binary structure;

when the second file corresponds to the stored first binary structure, determining if the second file is an updated version of the stored first binary structure by comparing a second filesize of the second file to the first filesize and comparing a second datestamp of the second file to the first datestamp; and

when the second filesize and the second datestamp do no match the first filesize and the first datestamp, respectively, deleting the stored first binary structure and the first filesize and the first datestamp, converting the second file to a second binary structure and storing the second filesize, the second datestamp and the second binary structure in the memory of the mobile device.